CS101- Algorithms and Programming I

Lab 02

Lab Objectives: Using Java to solve simple problems!

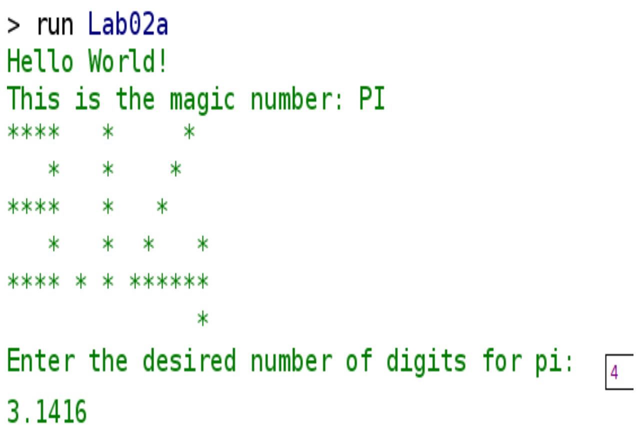
Notes:

- For all labs in CS 101, your solutions must conform to these CS101 style guidelines (rules!)
- Create a Lab02 workspace (i.e. the folder H:\private\cs101\lab02). This assignment has parts a, b, c, d, & e each of which should be placed in a separate project within the same Lab02 workspace. Note: only one project is active at a time. To work (Build/Run) a different project, right click on the project's name and select "Set as active project".
- Ideally, you should declare all constants first, followed by all the variables you will use, followed by the actual program. Whilst constants should be given values when declared, variables should not! Unfortunately, the textbook usually gives variables initial values when declaring them, and declares them in the middle of the program code, as needed. In general, we consider this bad practice and recommend you do not do it!
- These problems may be so simple you (think you) can do them in your head, however, it is a good idea to get into the habit of designing your program (on paper) first, before implementing it; doing so will save you a lot of time in the future when the problems get much larger.
- In all sample runs below; the user input is shown in **purple**.

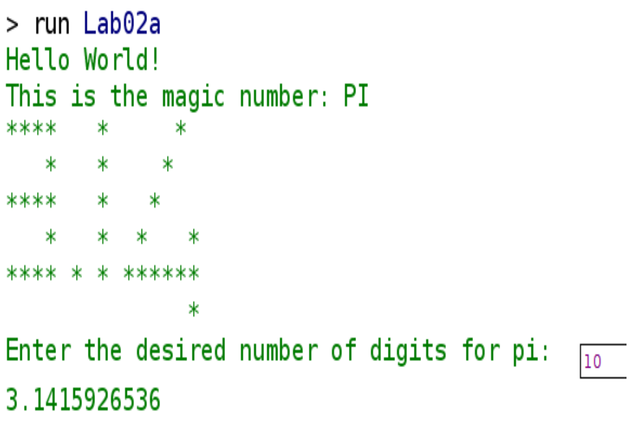
a. **“***Irrational but Well-Rounded”*

- You may start working on this lab assignment by watching the famous scene in Person of Interest. Pi is an irrational number, which means that it can’t be expressed as a fraction; it doesn’t end with a repeating pattern (like the decimal expression of 1/3, 0.33333…, in which the threes repeat forever), or terminate after a certain number of decimal places (like 3/4 or .75). It just keeps going, going, and going. So far, pi has been calculated to over 22 *trillion* digits. It took a computer with 24 hard drives, working nonstop for 105 days, to make that calculation. However, as some rookie programmers, let’s not get carried away immediately. We can control the number of decimal after the point up to some precision.
- You will demonstrate your ability to do basic output in Java using the println & printf methods in this question. Create a new project called Lab02a. First print “3.14” as an asterisk pattern as shown in the sample runs. Then take an integer input between 1-15 (inclusive) from the user, and that number will be the number of decimals after the point. Control the precision of Pi accordingly as in the sample runs.
- What happens if the input is bigger than 15? Why do you think that happens?

Sample Outputs:



```
> run Lab02a
Hello World!
This is the magic number: PI
****   *     *
   *   *    *
****   *   *
   *   *  *   *
**** * * ******
              *
Enter the desired number of digits for pi:  4
3.1416
```

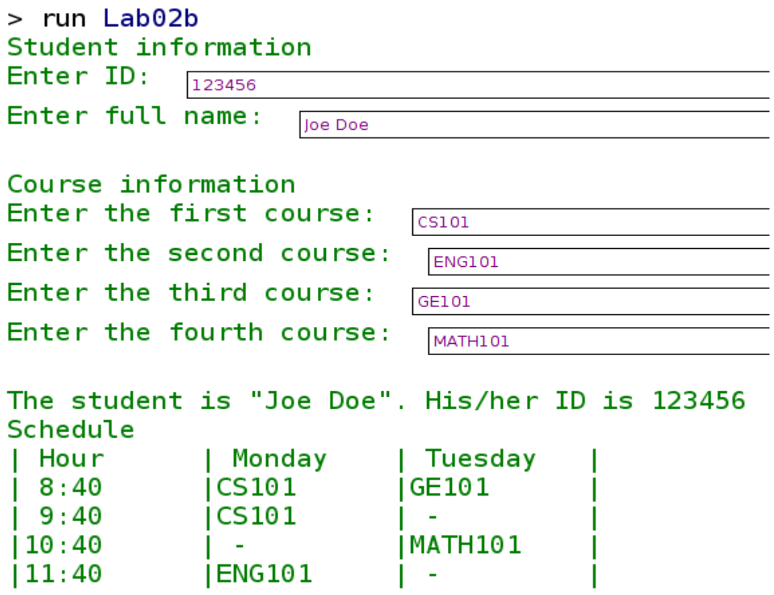


```
> run Lab02a
Hello World!
This is the magic number: PI
****   *     *
   *   *    *
****   *   *
   *   *  *   *
**** * * ******
              *
Enter the desired number of digits for pi:  10
3.1415926536
```

b. Practice input & output in Java. Create a new project called Lab02b..

You should use both input and output operations to construct a table showing a student's Monday and Tuesday morning class schedule. Examine the sample interaction below to design and implement your program. Collect all the relevant data from the user (note that the first two hours of Monday belong to the first course, the fourth hour, at 11:40, is for second course, and the third and fourth courses are at 8:40 and 10:40 on Tuesday). ***Hint:*** To get the output exactly right, including properly aligning table columns, you will need to use escape sequences to output characters such as new line, tab, slash, quotation mark, etc.

Sample run:

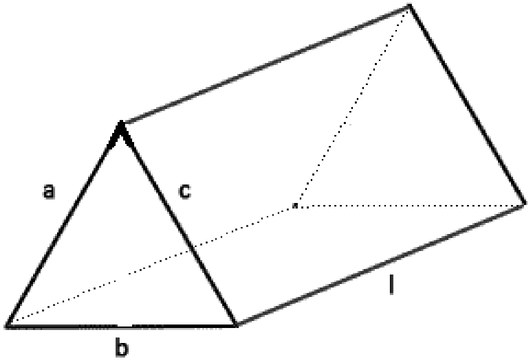


```
> run Lab02b
Student information
Enter ID:  123456
Enter full name:  Joe Doe

Course information
Enter the first course:  CS101
Enter the second course:  ENG101
Enter the third course:  GE101
Enter the fourth course:  MATH101

The student is "Joe Doe". His/her ID is 123456
Schedule
| Hour      | Monday    | Tuesday   |
| 8:40      |CS101      |GE101      |
| 9:40      |CS101      | -         |
|10:40      | -         |MATH101    |
|11:40      |ENG101     | -         |
```

c. Practice doing mathematical computations in Java. Create a new project called Lab02c. Your task is to find the volume of a triangular prism. Input the three sides of the triangular end of the prism (a, b, and c), and the length (l) of the prism from the user (all in centimeters). The area of a triangle whose lengths of all three sides can be calculated using the Equation 1 below. The volume of a triangular prism can be calculated using the Equation 2 below. When printing the area of the base triangle and the volume of the prism,, make sure that there are two and three decimals after the point, respectively. See a sample run below.



Equation 1: Heron's formula: the area, ***A***, of a triangle whose sides have lengths ***a***, ***b***, and ***c*** is

$$A = \sqrt{s(s-a)(s-b)(s-c)},$$

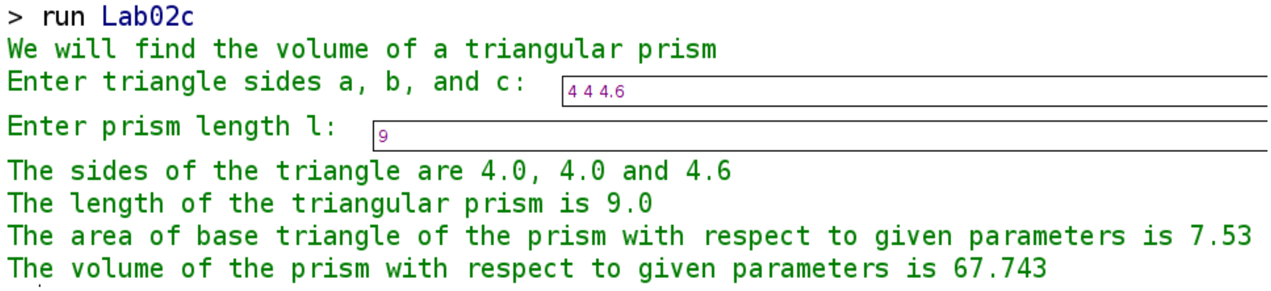
where **s** is the semi-perimeter of the triangle; that is,

$$s = \frac{a+b+c}{2}.$$

Equation 2: The volume, **V**, of a triangular prism whose length is **l** and area of base triangle is **A** is,

$$V = A \text{ x } l$$

Sample Run:

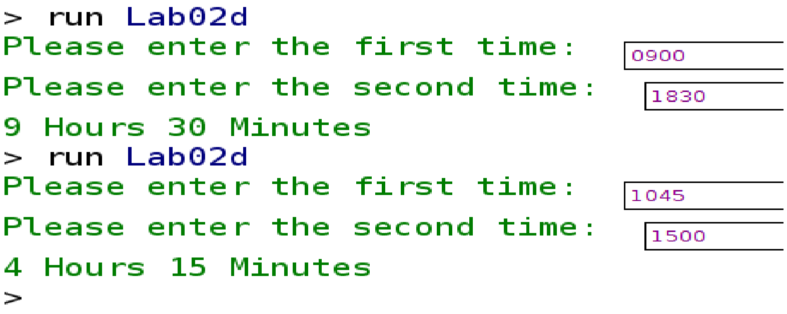


```
> run Lab02c
We will find the volume of a triangular prism
Enter triangle sides a, b, and c:  4 4 4.6
Enter prism length l:  9
The sides of the triangle are 4.0, 4.0 and 4.6
The length of the triangular prism is 9.0
The area of base triangle of the prism with respect to given parameters is 7.53
The volume of the prism with respect to given parameters is 67.743
```

d. Practice math operations in Java. Create a new project called Lab02d.

Write a program that takes two integer inputs for time from the user in the military format (i.e. 0900 for 9:00, 1830 for 18:30). And it should print the number of hours and minutes between the two times. Assume that the second time is later than the first one. See sample runs. ***Note:*** You should NOT read inputs as strings in this question. Assume that the user inputs are valid.

Sample runs:

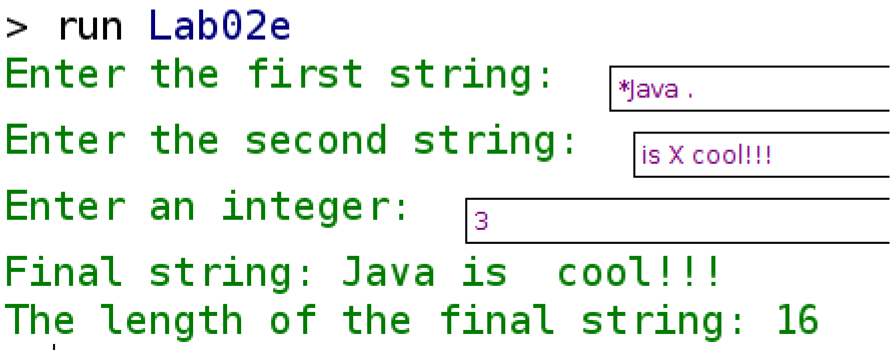


```
> run Lab02d
Please enter the first time:   0900
Please enter the second time:   1830
9 Hours 30 Minutes
> run Lab02d
Please enter the first time:   1045
Please enter the second time:   1500
4 Hours 15 Minutes
>
```

e. Practice doing String operations in Java. Create a new project called Lab02e.

Write a program that reads two string inputs and one integer input that will correspond to the index in the second string. Firstly, your program should create a version of the first input string without the first and last chars. Secondly, it should create a version of the second string without the char that was in the index that you took as an input previously. You can assume that the length of the first string will be at least 2 and integer input is valid. Then you should print the combination of the latest versions of two strings. You also should print the length of the final string. See a sample run.

Sample run:



```
> run Lab02e
Enter the first string:   *Java .
Enter the second string:   is X cool!!!
Enter an integer:   3
Final string: Java is  cool!!!
The length of the final string: 16
```